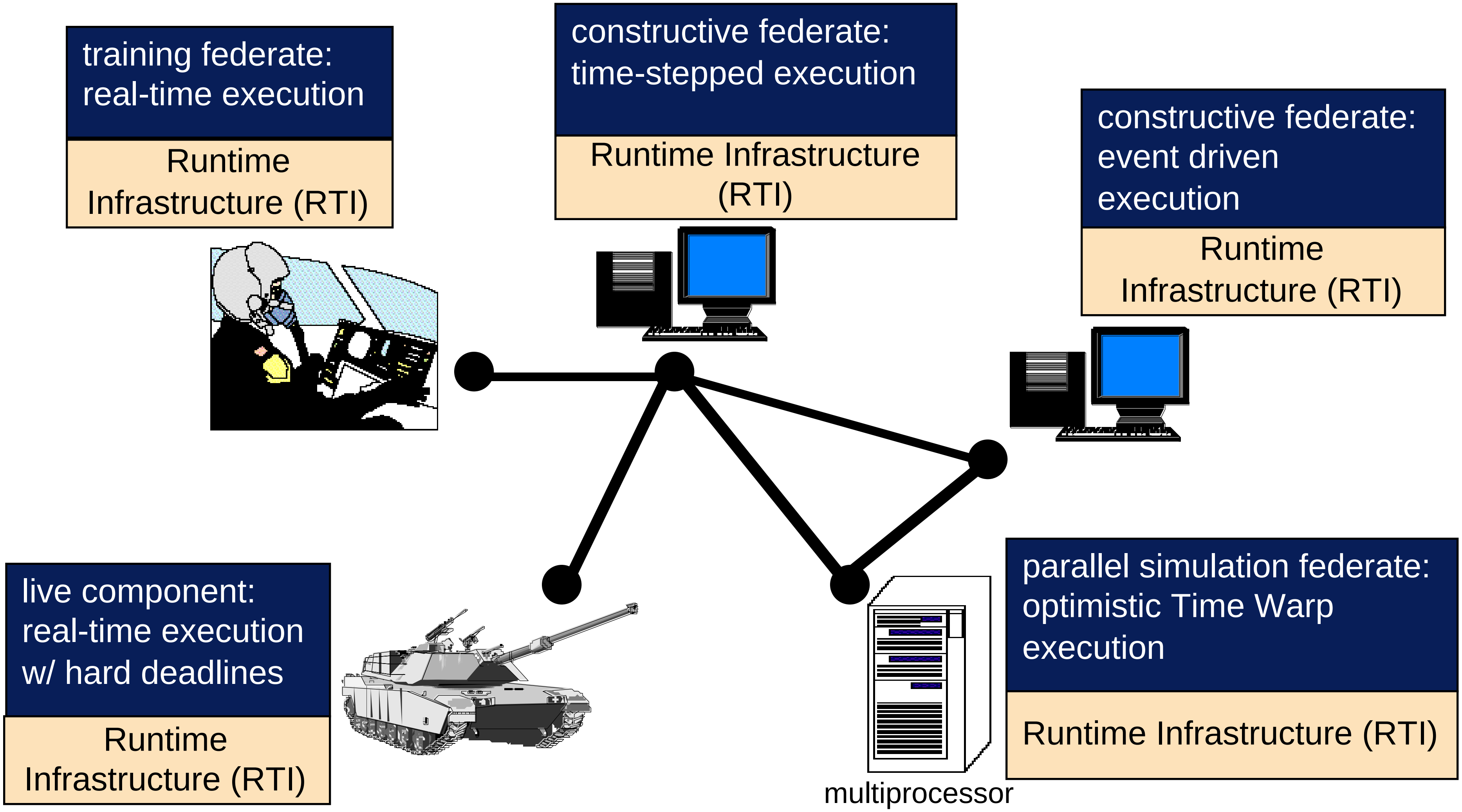
Time Management in the HLA
Baseline Definition

Richard M. Fujimoto
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
fujimoto@cc.gatech.edu

Challenge: Time Management Interoperability



training federate:
real-time execution

Runtime Infrastructure (RTI)

constructive federate:
time-stepped execution

Runtime Infrastructure (RTI)

constructive federate:
event driven execution

Runtime Infrastructure (RTI)

live component:
real-time execution w/ hard deadlines

Runtime Infrastructure (RTI)

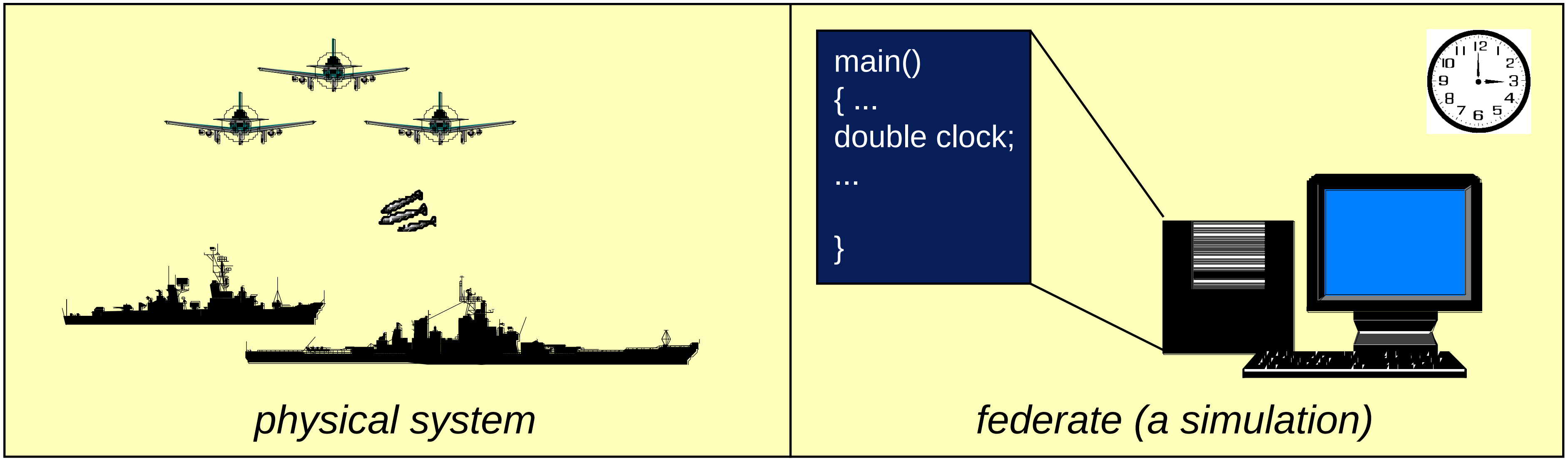
parallel simulation federate:
optimistic Time Warp execution

Runtime Infrastructure (RTI)

Goal: provide services to support interoperability among federates with different local time management schemes in a single federation execution.

Time

- ***physical system***: the actual or imagined system being modeled
- ***simulation***: a system that emulates the behavior of a physical system



```
main()
{ ...
double clock;
...

}
```

- ***physical time:*** time in the physical system (e.g., 0000 to 1700 hours, December 7, 1941)
- ***federation time axis (FTA):*** a totally ordered sequence of values representing physical time (floating point value ranging from 0.0 to 17.0)
- ***federate time:*** current point on FTA perceived by a specific federate (e.g., 4.0)
- ***wallclock time:*** time during the execution of the simulation, usually output from a hardware clock (e.g., 0900 to 1000 hours on September 16, 1996)

Time Management

Time management is concerned with the mechanisms used by federates to advance along the federation time axis

paced (constrained) vs. unpaced (unconstrained) execution

- ***(scaled) real-time execution***: each federate is paced so federate time advances at a rate S times faster or slower than wallclock time
- ***as-fast-as-possible execution***: no fixed relationship exists between advances in federate and wallclock time

independent vs. coordinated time advances

- ***independent*** **(e.g., training): each federate advances its federate time independently of other federates**

  federate time same as wallclock time (scaled, plus an offset)

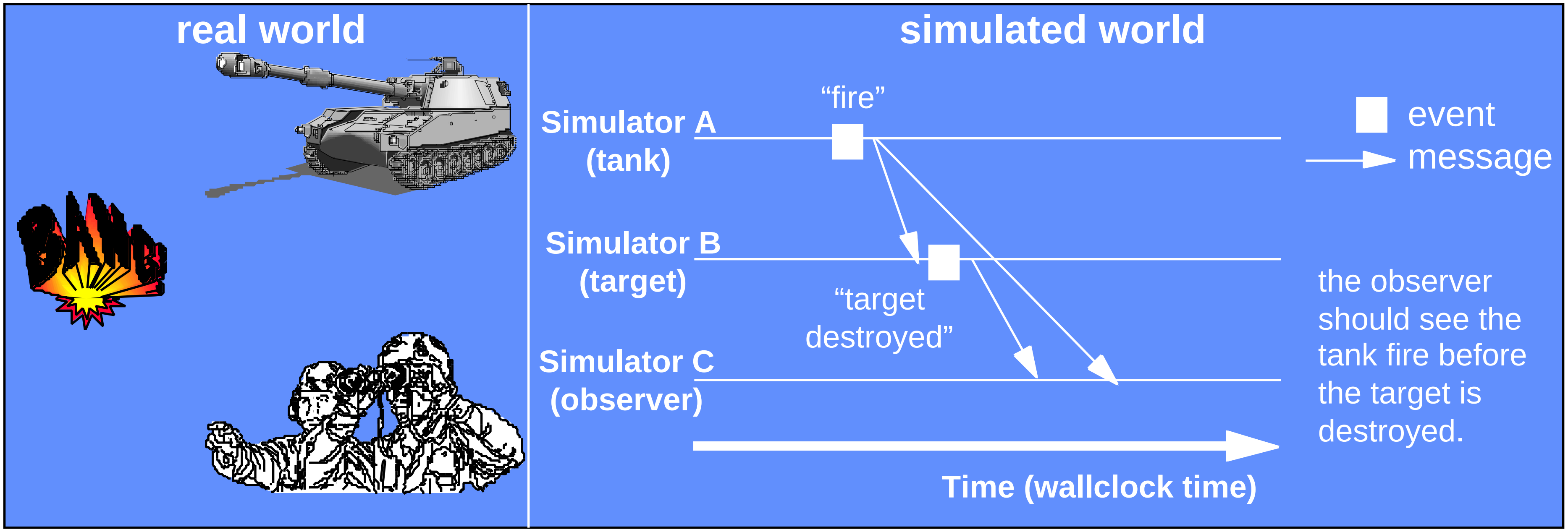
- ***coordinated*** **(e.g., analysis): time advances are coordinated to ensure the federation preserves before/after relationships in the physical system**

  federate time same as logical time

example	**paced?**	**time advance**
DIS training exercise	*yes*	*independent*
ALSP (as-fast-as-possible)	*no*	*coordinated*
ALSP (human-in-the-loop)	*yes*	*coordinated*

Causality

- “Things” happen in the real world in a certain order (e.g., cause & effect).
- It should appear that events in the simulated world happen in the same order as the real world actions that they represent.



the observer should see the tank fire before the target is destroyed.

Causality: If event A *“happens before”* event B, the message for A should be delivered before the message for B

If the message for the “fire” event is delayed in the network, the observer will “see” the target is destroyed before the tank fired upon it!

Temporal anomalies such as this may or may not be a problem, depending on the federation’s goals

Message Ordering Services

The baseline HLA provides two types of message ordering:

- ***receive order:*** messages passed to federate in order of reception
- ***time stamp order (TSO):*** successive messages passed to federate have non-decreading time stamps

Property	Receive Order	Time Stamp Order (TSO)
Latency	*low*	*higher*
reproduce before and after relationships?	*no*	*yes*
all federates see same ordering of events?	*no*	*yes*
execution repeatable?	*no*	*yes*
typical applications	*training, T&E*	*analysis*

- receive order minimzes latency, does not prevent temporal anomalies
- TSO prevents temporal anomalies, but has somewhat higher latency

Logical Time

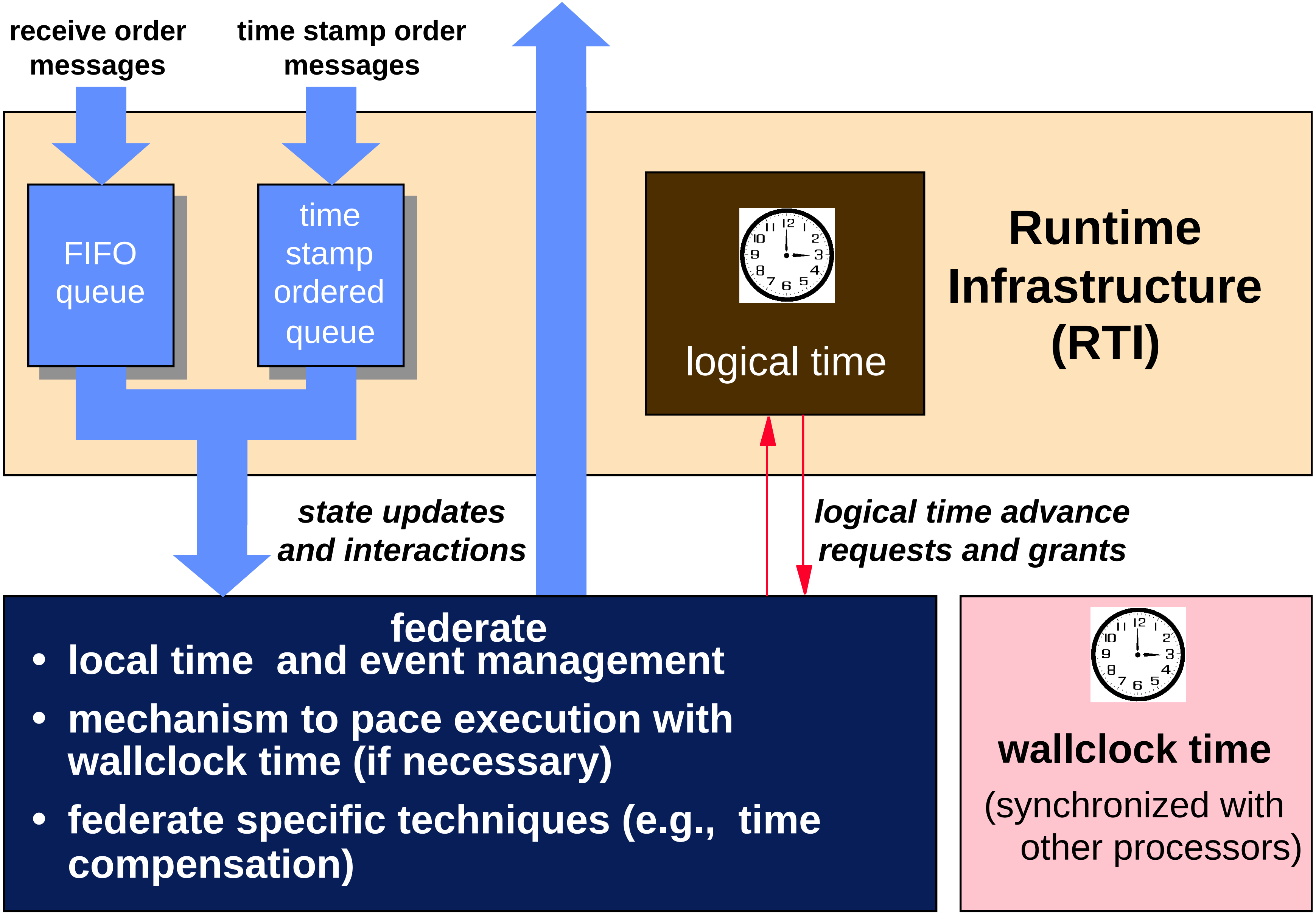
Classical discrete event simulation programs process all events in time stamp order

A mechanism is required to enable federates to interleave processing of local events with those received from other federates

Logical time:

- **if the logical time of a federate is T, the RTI guarantees no more TSO messages will be passed to the federate with time stamp < T**
- **local events with time stamp < T can be “safely” processed**
- **logical time in a federate only advances when that federate explicitly requests an advance:**
  - **`Time Advance Request (T)`: requests advance to time T**
  - **`Next Event Request (T)`: requests advance to time stamp of next TSO message, or T, which ever is smaller**
  - **RTI issues `Time Advance Grant(T′)` when logical time advanced to T’**
- **applies only to federates utilizing time stamp order (e.g., analysis)**
- **federates responsible for pacing logical time advances with wallclock time in (scaled) real-time federations**

Time Management Services



receive order messages

time stamp order messages

FIFO queue

time stamp ordered queue

logical time

Runtime Infrastructure (RTI)

state updates and interactions

logical time advance requests and grants

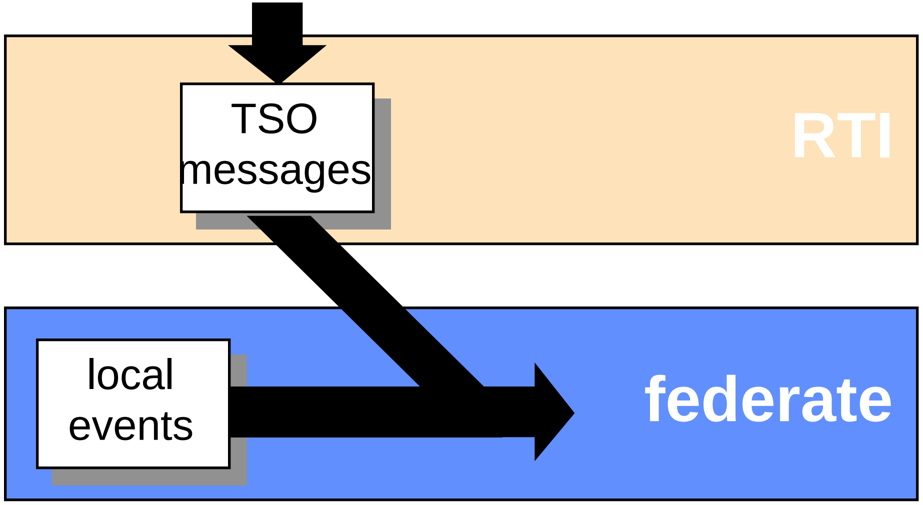
federate

- **local time and event management**
- **mechanism to pace execution with wallclock time (if necessary)**
- **federate specific techniques (e.g., time compensation)**

wallclock time

(synchronized with other processors)

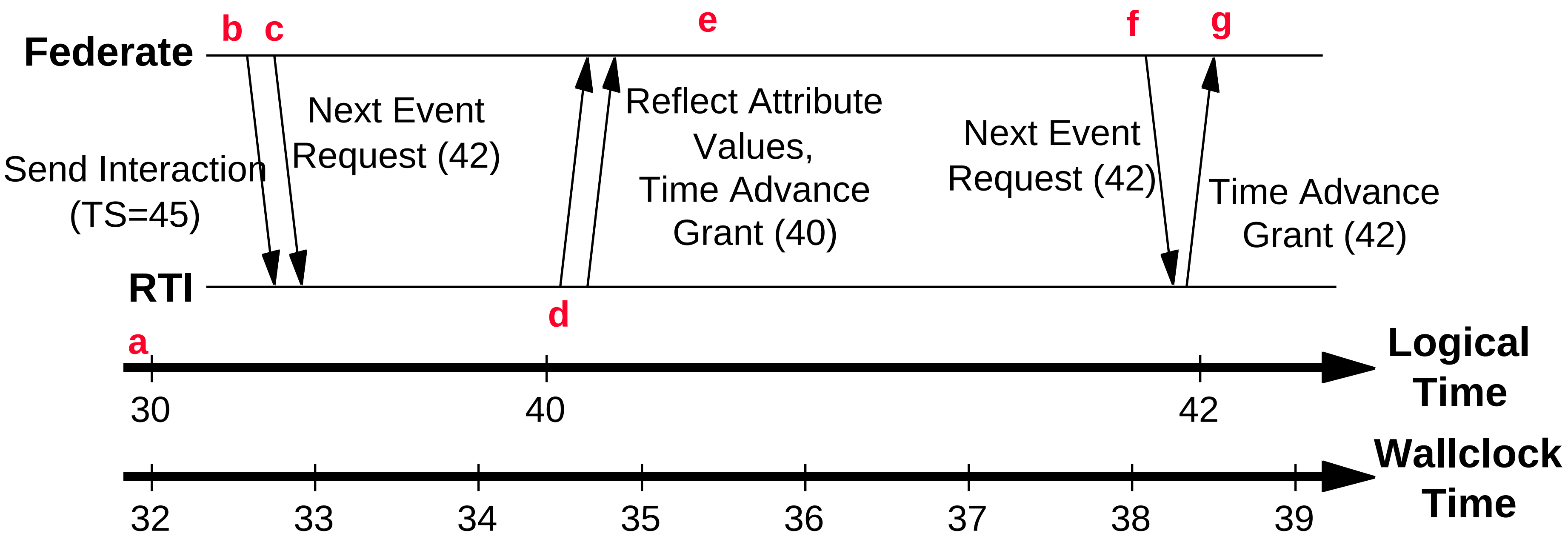
Example: Event Driven Federate



Goal: merge TSO messages (events from other federates) with local events so all events are processed in time stamp order

```
/* now is a local variable tracking the logical time of the federate */
while (simulation still in progress)
   TSlocal = time stamp of next local event notice
   invoke Next Event Request (TSlocal) service
            Receive state updates and interactions, send new updates and interactions
            honor RTI service request for Time Advance Grant
   if (no TSO message(s) received in above RTI service request)
            now = TSlocal
            process the next local event notice send new updates and interactions
   else
            now = time stamp of last TSO message delivered to federate
```

Scenario: Event Driven Federate



(a) logical time initially 30, one message in TSO queue with time stamp 40

(b) federate: send interaction message after processing local event

(c) federate: next local event has time stamp 42, request next event w/ TS < 42

(d) RTI: determines no messages with TS < 40 forthcoming, deliver TS=40 message and time advance grant, logical time is advanced to 40

(e) federate: processes time stamp 40 event

(f) federate: next local event has time stamp 42 (still), request next event w/ TS < 42

(g) RTI: determines no messages w/ TS<42 forthcoming, grant time advance to 42, process local event

Other Aspects of Time Management Services

- Utilizing time stamp order; each federate declares itself to be:
  - logical time constrained: may receive TSO messages (RTI must control logical time advances by this federate, or
  - logical time regulating: may send TSO messages (RTI must consider this federate in determining what TSO messages can be delivered to a federate), or
  - neither (e.g., DIS training federate), or
  - both (e.g., analysis federate)
- Federates using logical time must specify lookahead
  - constraint: the time stamp of an event generated by the federate must be at least the federate’s current time + lookahead
  - can significantly improve performance of the federation execution

Optimistic Time Management Services

Mechanisms to ensure events are processed in time stamp order:

- conservative: protocols to *avoid* out of order event processing (e.g., CMB)
- optimistic: *detect* out-of-order event processing, *recover* (e.g., Time Warp)

Requirements

- support federations including conservative and/or optimistic federates
- federates not aware of local time management mechanism of other federates (optimistic or conservative)
- optimistic events (events that may be later canceled) cannot be delivered to conservative federates that cannot roll back
- optimistic events should be delivered to other optimistic federates
- individual federates may be sequential or parallel simulations

HLA time management services for optimistic federates:

- early release of messages to optimistic federates **(`Flush Queue Request`** and **`Grant`)**
- cancelation of previously sent messages: anti-messages **(`Retract`)**
- provide sufficient information to optimistic federate to compute *Global Virtual Time* locally **(LBTS)**